Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Отчет о лабораторной работе № 6

по дисциплине «Программирование на языке JAVA»

Вариант № 5

Выполнили: ст-ты гр. 19ВВ2

Горбунов Д.А.

Кобзев М.И.

Муромский Д.А.

Проверили:

Юрова О.В.

Карамышева Н.С.

2022

**Цель работы:** научиться создавать клиент-серверные приложения c использованием стандартных классов Java.

**Задание на лабораторную работу:** Модифицировать приложение из предыдущей лабораторной работы, реализовав клиент-серверную архитектуру, обеспечивающую распределенное вычисление определенного интеграла на нескольких вычислительных узлах (клиентах) при этом каждый узел использует несколько нитей, как в предыдущей работе. Сервер не занимается вычислениями, а лишь реализует взаимодействие с пользователем и агрегацию результатов вычислений от клиентов. Нечетные варианты используют протокол UDP, а четные TCP.

**Решение:**

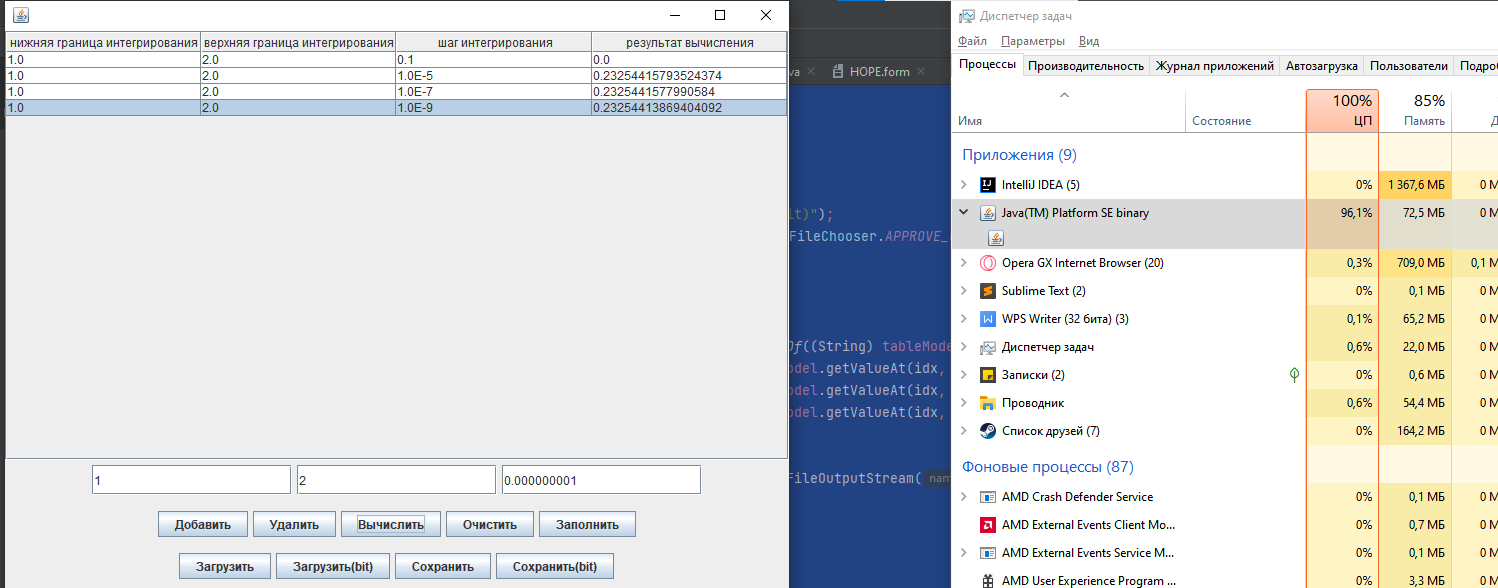


Рисунок 1 - Загрузка ЦП на 100% (5 потоков на 4 ядра)

**Вывод:** научились создавать клиент-серверные приложения c использованием стандартных классов Java.

**Листинг Main.java**

package com.company;  
  
  
import java.util.concurrent.Delayed;  
  
public class Main {  
  
 public static void main(String[] args) {  
 System.out.println("Hello World!");  
 HOPE w = new HOPE();  
 w.setVisible(true);  
 }  
}

**Листинг MyExeption.java**

package com.company;  
  
import javax.swing.\*;  
  
public class MyExeption extends Exception{  
 public MyExeption(String str){  
 JOptionPane.showMessageDialog(null, str, "Error 404", JOptionPane.ERROR\_MESSAGE);  
 }  
}

**Листинг MyThread.java**

package com.company;  
  
public class MyThread extends Thread{  
 private double start;  
 private double end;  
 private double step;  
 private double result = 0.0;  
  
 private double E = 2.7182818284590452353602874713527;  
  
 public MyThread(double \_start, double \_end, double \_step){  
 start = \_start;  
 end = \_end;  
 step = \_step;  
 }  
  
 @Override  
 public void run(){  
 while(start < end){  
 if(start + step > end){  
 step = end - start;  
 }  
 result += 0.5 \* (Math.pow(E, -start) + Math.pow(E, -(start + step))) \* step;  
 start += step;  
 }  
 }  
  
 public synchronized double getResult(){  
 return result;  
 }  
}

**Листинг RecIntegral.java**

package com.company;  
  
import java.io.Serializable;  
  
public class RecIntegral implements Serializable {  
 public double start = 0.0;  
 public double end = 0.0;  
 public double step = 0.0;  
 public double result = 0.0;  
  
 public RecIntegral(){  
  
 }  
  
 public RecIntegral(double \_start, double \_end, double \_step) throws MyExeption {  
 try {  
 Rec(\_start, \_end, \_step, 0.0);  
 }catch (MyExeption myEx){  
 throw new MyExeption(myEx.getMessage());  
 }  
 }  
  
  
 public RecIntegral(double \_start, double \_end, double \_step, double \_result) throws MyExeption {  
 try {  
 Rec(\_start, \_end, \_step, \_result);  
 }catch (MyExeption myEx){  
 throw new MyExeption(myEx.getMessage());  
 }  
 }  
  
  
 public void Rec(double \_start, double \_end, double \_step, double \_result) throws MyExeption {  
 if(CheckMaxMin(\_start, 0.000001, 1000000) &&  
 CheckMaxMin(\_start, 0.000001, 1000000)&&  
 CheckMaxMin(\_start, 0.000001, 1000000)){  
 start = \_start;  
 end = \_end;  
 step = \_step;  
 result = \_result;  
 }else{  
 throw new MyExeption("Number out of range!");  
 }  
 }  
  
 private boolean CheckMaxMin(double number, double min, double max){  
 if(number > min && number < max){  
 return true;  
 }  
 return false;  
 }  
}

**Листинг HOPE.java**

package com.company;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.LinkedList;  
import javax.swing.filechooser.FileNameExtensionFilter;  
import java.util.Vector;  
  
public class HOPE extends JFrame{  
 private DefaultTableModel tableModel;  
 private JTable table1;  
  
 private JTextField textField1;  
 private JTextField textField2;  
 private JTextField textField3;  
 private JButton add;  
 private JButton remove;  
 private JButton calc;  
 private JButton clear;  
 private JButton fill;  
 private JButton load;  
 private JButton load\_bit;  
 private JButton save;  
 private JButton save\_bit;  
  
 private double E = 2.7182818284590452353602874713527;  
 private ArrayList<RecIntegral> ne\_chet = new ArrayList<>();  
 private LinkedList<RecIntegral> chet = new LinkedList<>();  
 private JFileChooser fileChooser;  
  
 Object[] columnNames = {"нижняя граница интегрирования",  
 "верхняя граница интегрирования",  
 "шаг интегрирования",  
 "результат вычисления"};  
  
  
 public static String State(Thread.State ts) {  
 if (ts == Thread.State.BLOCKED)  
 return "BLOCKED";  
 if (ts == Thread.State.NEW)  
 return "NEW";  
 if (ts == Thread.State.RUNNABLE)  
 return "RUNNABLE";  
 if (ts == Thread.State.TIMED\_WAITING)  
 return "TIMED\_WAITING";  
 return "WAITING";  
 }  
  
  
 public HOPE(){  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
  
 fileChooser = new JFileChooser();  
 fileChooser.setFileSelectionMode(JFileChooser.FILES\_ONLY);  
 fileChooser.setFileFilter(new FileNameExtensionFilter("\*.ser", "\*"));  
  
 CreateTextModel();  
 CreateTextField();  
 CreateButton();  
  
 CreateForm();  
 }  
  
 private void CreateTextModel(){  
 tableModel = new DefaultTableModel();  
 tableModel.setColumnIdentifiers(columnNames);  
  
 table1 = new JTable(tableModel);  
 }  
  
 private void CreateTextField(){  
 textField1 = new JTextField();  
 textField1.setPreferredSize(new Dimension(200, 30));  
 textField2 = new JTextField();  
 textField2.setPreferredSize(new Dimension(200, 30));  
 textField3 = new JTextField();  
 textField3.setPreferredSize(new Dimension(200, 30));  
 }  
  
 private void CreateButton(){  
 add = new JButton("Добавить");  
 add.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 double[] data;  
 try {  
 data = new double[]{Double.valueOf((String) textField1.getText()),  
 Double.valueOf((String) textField2.getText()),  
 Double.valueOf((String) textField3.getText())};  
 }catch(Throwable t){  
 return;  
 }  
  
 if (ne\_chet.size() >= chet.size()){  
 try {  
 chet.add(new RecIntegral(data[0],  
 data[1], data[2]));  
 }catch (MyExeption myExeption){  
 return;  
 }  
  
 tableModel.insertRow(tableModel.getRowCount(), new String[] {  
 String.valueOf(chet.getLast().start), String.valueOf(chet.getLast().end),  
 String.valueOf(chet.getLast().step), String.valueOf(chet.getLast().result)});  
 }else{  
 try {  
 ne\_chet.add(new RecIntegral(data[0],  
 data[1], data[2]));  
 }catch (MyExeption myExeption){  
 return;  
 }  
  
 tableModel.insertRow(tableModel.getRowCount(), new String[] {  
 String.valueOf(ne\_chet.get(ne\_chet.size() - 1).start),  
 String.valueOf(ne\_chet.get(ne\_chet.size() - 1).end),  
 String.valueOf(ne\_chet.get(ne\_chet.size() - 1).step),  
 String.valueOf(ne\_chet.get(ne\_chet.size() - 1).result)});  
 }  
 }  
 });  
  
 remove = new JButton("Удалить");  
 remove.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1) {  
 return;  
 }  
 tableModel.removeRow(idx);  
 if(idx % 2 == 0){  
 chet.remove(idx / 2);  
 }else{  
 ne\_chet.remove(idx / 2);  
 }  
 }  
 });  
  
 calc = new JButton("Вычислить");  
 calc.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1){  
 return;  
 }  
 double[] data;  
 try {  
 data = new double[]{Double.valueOf((String) tableModel.getValueAt(idx, 0)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 1)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 2))};  
 }catch(Throwable t){  
 tableModel.setValueAt("NULL", idx, 3);  
 return;  
 }  
 double start, end, step, result;  
 System.out.println();  
 start = data[0];  
 end = data[1];  
 step = data[2];  
 result = 0.0;  
  
 double stepThread = (end - start) / 5;  
 MyThread[] myThread = new MyThread[5];  
 try {  
 for (int i = 0; i < 5; i++) {  
 myThread[i] = new MyThread(start + i \* stepThread, end - (4 - i) \* stepThread, step);  
 myThread[i].start();  
 }  
 for(int i = 0; i < 5; i++){  
 myThread[i].join();  
 }  
  
 for(int i = 0; i < 5; i++){  
 result += myThread[i].getResult();  
 }  
 }catch(NullPointerException ex){  
 System.out.println("Error" + ex.getMessage());  
 return;  
 }catch (InterruptedException ex){  
  
 }  
  
 tableModel.setValueAt(result, idx, 3);  
 }  
 });  
  
 clear = new JButton("Очистить");  
 clear.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 while(tableModel.getRowCount() > 0){  
 tableModel.removeRow(0);  
 }  
 }  
 });  
  
 fill = new JButton("Заполнить");  
 fill.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int i = 0;  
 while(i <= chet.size() + ne\_chet.size()){  
 if(i % 2 == 0) {  
 if ((i / 2) < chet.size()) {  
 tableModel.insertRow(i, new String[]{  
 String.valueOf(chet.get(i / 2).start), String.valueOf(chet.get(i / 2).end),  
 String.valueOf(chet.get(i / 2).step), String.valueOf(chet.get(i / 2).result)});  
 }  
 }else {  
 if((i / 2) < ne\_chet.size() ) {  
 tableModel.insertRow(i, new String[]{  
 String.valueOf(ne\_chet.get(i / 2).start), String.valueOf(ne\_chet.get(i / 2).end),  
 String.valueOf(ne\_chet.get(i / 2).step), String.valueOf(ne\_chet.get(i / 2).result)});  
 }  
 }  
 i++;  
 }  
 }  
 });  
  
 ///////////////////////////////////////////////////////////////////////////////  
  
 load\_bit = new JButton("Загрузить(bit)");  
 load\_bit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e){  
 fileChooser.setDialogTitle("Выборите файл");  
 int result = fileChooser.showOpenDialog(HOPE.this);  
 if(result == JFileChooser.APPROVE\_OPTION){  
 File fileReader = fileChooser.getSelectedFile();  
 ObjectInputStream obj\_in = null;  
 RecIntegral restObj = null;  
 try{  
 obj\_in = new ObjectInputStream(new BufferedInputStream(new FileInputStream( fileReader.getAbsolutePath())));  
 restObj = (RecIntegral)obj\_in.readObject();  
 }catch (IOException ex){  
 ex.printStackTrace();  
 }catch (ClassNotFoundException exception){  
  
 }  
 tableModel.insertRow(tableModel.getRowCount(), new String[] {  
 String.valueOf(restObj.start), String.valueOf(restObj.end),  
 String.valueOf(restObj.step), String.valueOf(restObj.result)});  
 }  
 }  
 });  
  
 load = new JButton("Загрузить");  
 load.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 fileChooser.setDialogTitle("Выберите файл");  
 if(fileChooser.showOpenDialog(null) == JFileChooser.APPROVE\_OPTION){  
 String file = fileChooser.getSelectedFile().toString();  
  
 try{  
 BufferedReader in = new BufferedReader(new FileReader(file));  
 String[] s = new String[4];  
 try {  
 for(int i = 0; i < 4; i++){  
 s[i] = in.readLine();  
 }  
 tableModel.insertRow(tableModel.getRowCount(), s);  
 }finally {  
 in.close();  
 }  
 }catch (IOException ex){  
 throw new RuntimeException(ex);  
 }  
 }  
  
  
 }  
 });  
  
 save\_bit = new JButton("Сохранить(bit)");  
 save\_bit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1) {  
 return;  
 }  
 fileChooser.setDialogTitle("Сохранение файла(bit)");  
 if(fileChooser.showSaveDialog(null) == JFileChooser.APPROVE\_OPTION){  
 try{  
 ObjectOutputStream obj\_out = null;  
 RecIntegral restObj;  
  
 restObj = new RecIntegral(Double.valueOf((String) tableModel.getValueAt(idx, 0)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 1)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 2)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 3)));  
  
  
 obj\_out = new ObjectOutputStream(new FileOutputStream(fileChooser.getSelectedFile() + ".ser"));  
 obj\_out.writeObject(restObj);  
 }catch (IOException ex){  
 ex.printStackTrace();  
 }catch(MyExeption ex){  
 return;  
 }  
 }  
 }  
 });  
  
 save = new JButton("Сохранить");  
 save.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1) {  
 return;  
 }  
  
 fileChooser.setDialogTitle("Сохранение файла");  
 if(fileChooser.showSaveDialog(null) == JFileChooser.APPROVE\_OPTION){  
 File file = new File(fileChooser.getSelectedFile() + ".ser");  
 try{  
 if(!file.exists()){  
 file.createNewFile();  
 }  
  
 PrintWriter out = new PrintWriter(file);  
 try {  
 for(int i = 0; i < 4; i++)  
 out.println(tableModel.getValueAt(idx, i));  
 }finally {  
 out.close();  
 }  
 }catch (IOException ex){  
 throw new RuntimeException(ex);  
 }  
 }  
 }  
 });  
 }  
  
 private void CreateForm(){  
 Box contents = new Box(BoxLayout.Y\_AXIS);  
 contents.add(new JScrollPane(table1));  
  
 JPanel text = new JPanel();  
 text.add(textField1);  
 text.add(textField2);  
 text.add(textField3);  
 contents.add(text);  
  
 JPanel button = new JPanel();  
 button.add(add);  
 button.add(remove);  
 button.add(calc);  
 button.add(clear);  
 button.add(fill);  
 contents.add(button);  
  
 JPanel button\_load\_save = new JPanel();  
 button\_load\_save.add(load);  
 button\_load\_save.add(load\_bit);  
 button\_load\_save.add(save);  
 button\_load\_save.add(save\_bit);  
 contents.add(button\_load\_save);  
  
 getContentPane().add(contents);  
  
 setSize(800, 600);  
 setVisible(true);  
 }  
}